Jurnal Elektronik Ilmu Komputer Udayana
Volume 13, No 2. November 2024

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Sistem Digital Signature Untuk Verifikasi Dokumen Digital Menggunakan Mac Address Pada SHA-256 dan AES-128

I Dewa Gde Putra Anga Biara[a1], I Putu Gede Hendra Suputra[a2], I Gusti Agung Gede Arya Kadyanan[a3], I Gede Arta Wibawa[a4]
[a]Program Studi Informatika, Fakultas Matermatika dan Ilmu Pengetahuan Alam
Universitas Udayana
Jalan Raya Kampus Unud, Badung, 08361, Bali, Indonesia
[1]angabiara@gmail.com
[2]hendra.suputra@unud.ac.id
[3]gungde@unud.ac.id
[4]gede.arta@unud.ac.id

Abstract

The advancement of internet technology has drastically changed the way we communicate and share information, enabling the transmission of data in various formats such as audio, video, images, and files. However, this convenience also brings new challenges such as document forgery and the low authenticity of files. This research aims to develop a desktop application-based system capable of verifying the authenticity of digital documents by implementing digital signatures using MAC addresses on the AES-128 and SHA-256 cryptographic algorithms. The system is designed to perform document hashing and encryption processes, store encryption results in a database, and provide key recipients with access for decryption. The results of the research indicate that the system can perform encryption and decryption very well, as evidenced by the successful decryption of ciphertext and the retrieval of the original document verified by the user. Additionally, the system's performance was tested using the avalanche effect method, which showed an average of 50.94%, indicating a high level of security that is difficult for attackers to breach. Black-Box testing involving login, registration, document signing, and document verification scenarios also demonstrated that the system functions according to specifications and can handle core operations effectively. The system's success in ensuring the authenticity and integrity of digital documents provides an effective solution to the growing problem of document forgery in the digital age.

Keywords: *cryptography, AES-128, SHA-256, encryption, decryption*

1. Pendahuluan

Teknologi saat ini tengah berkembang dengan sangat pesat. Dengan adanya internet menjadikan segala sesuatu yang kita lakukan sangat bergantung terhadap internet. Salah satu hal yang terdampak dengan adanya internet adalah cara kita berkomunikasi. Selain untuk mengirimkan pesan, dengan adanya internet kini jenis informasi yang bisa dikirimkan menjadi lebih banyak, sebagai contoh informasi itu dapat berupa suara, video, gambar maupun file. Pada riset yang dilakukan pada tahun 2022 oleh Petroc Taylor, terdapat 3 aplikasi untuk berbagi file yang menjadi peringkat teratas yang digunakan secara global yaitu Google Drive memimpin dengan pangsa pasar sebesar 28,13 persen, disusul oleh Dropbox yang memiliki basis di San Francisco dengan memiliki pangsa pasar sebesar 27,6 persen, dan yang terakhir adalah Box dengan menguasai pangsa pasar 14,5 persen. Meskipun dengan adanya aplikasi yang mempermudah untuk melakukan pengiriman file melalui internet, timbul masalah baru yaitu penyebaran file yang tidak benar dan rendahnya tingkat keaslian dari sebuah file. Maka dari itu kebutuhan akan keamanan serta verifikasi keaslian data maupun informasi semakin meningkat.

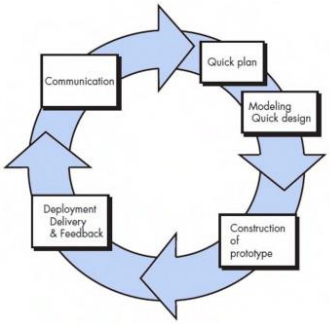
Salah satu dampak negatif dari mudahnya akses untuk menyebarkan file adalah pemalsuan file. Menurut artikel dari *pusiknas.polri.go.id*, terdapat 95 data penindakan kasus pemalsuan surat dari tanggal 1 – 12 Januari 2022. Dari data yang telah didapat tersebut dapat dirata-ratakan untuk setiap harinya terdapat 7 kasus pemalsuan dokumen dan surat. Hal ini menunjukkan bahwa tinggi nya angka kasus pemalsuan dokumen dan surat yang masih terjadi saat ini. Untuk dapat menjaga keaslian dari suatu dokumen, perlu dilakukannya pengecekan pada dokumen yang telah diterbitkan untuk mengecek keaslian dari dokumen tersebut. Memberikan digital signature kepada dokumen yang akan diterbitkan merupakan salah satu cara untuk menjaga keaslian dokumen tersebut. Tanda tangan digital adalah sebuah nomor yang bergantung pada beberapa rahasia yang hanya diketahui oleh si penandatangan (kunci pribadi si penandatangan), dan juga pada isi pesan yang ditandatangani [1].

Kriptografi merupakan metode pengamanan data dengan mengubah pesan menjadi bentuk lain yang tidak dapat dipahami tanpa kunci yang tepat, digunakan untuk mengamankan file seperti dokumen, video, gambar, dan audio [2]. Kriptografi dibagi menjadi dua jenis utama: simetris dan asimetris. Kriptografi simetris menggunakan kunci privat yang sama untuk enkripsi dan dekripsi, dengan metode seperti DES, RC4, AES, OTP, dan Blowfish. Sementara itu, kriptografi asimetris menggunakan kunci privat dan publik, dengan algoritma seperti RSA, El Gamal, Elliptic Curve, Hill Cipher, dan Diffie-Hellman. AES, sebagai salah satu metode enkripsi simetris, mengubah data dalam blok 128 bit dan memiliki varian kunci seperti AES-128, AES-192, dan AES-256. Keamanan AES bergantung pada kerahasiaan kunci yang digunakan untuk enkripsi dan dekripsi.

Penelitian sebelumnya menunjukkan penerapan AES-128 untuk pengamanan data kependudukan di Dinas Dukcapil Pematangsiantar, menghasilkan file yang tidak dapat dipahami oleh pihak ketiga dan mengamankan berbagai jenis file seperti doc, xls, ppt, pdf, jpg, dan png [3]. Penelitian lain mengkaji penerapan digital signature menggunakan MAC address dengan AES-128 dan SHA-2 256 bit, yang meningkatkan keamanan tanda tangan digital karena keunikan MAC address dan menghasilkan string acak yang unik dengan kecepatan yang normal. Berdasarkan penelitian ini, penulis merancang aplikasi desktop untuk verifikasi keaslian dokumen digital menggunakan MAC address pada algoritma AES-128 dan SHA-256.

2. Metode Penelitian

Metode yang digunakan dalam membangun sistem ini adalah metode prototyping. Model Prototyping merupakan suatu metode pengembangan perangkat lunak dimana prototipe dibangun, diuji, dan dikembangkan kembali hingga mencapai prototipe yang dapat diterima seperti yang ditunjukkan pada Gambar 1. Pengembangan dengan model prototipe cocok digunakan dalam situasi dimana persyaratan proyek tidak diketahui secara rinci. Dalam tahap komunikasi, dilakukan wawancara dan observasi dengan pemilik perusahaan untuk mengumpulkan kebutuhan dan masalah pengguna. Hal ini bertujuan untuk memperoleh gambaran umum mengenai sistem yang akan dibuat. Tahap *Quick Plan and Modeling Quick Design* difokuskan pada aspek tampilan atau antarmuka sistem, termasuk keluaran atau output yang diberikan kepada pengguna. Pada tahap ini dilakukan proses desain rancangan sistem dengan menggunakan Unified Modeling Language (UML), diantaranya user case dan activity diagram dan Entity Relationship Diagram (ERD). Melakukan konstruksi atau pembangunan kerangka atau desain awal dari perangkat lunak yang akan dikembangkan. Pada perancangan aplikasi bahasa pemrograman yang akan digunakan adalah Python untuk Front-End dan Golang untuk Back-End dengan tools Visual Studio Code untuk pengembangan aplikasi. Aplikasi yang dirancang menggunakan MySQL untuk penyimpanan data hasil signature. Melakukan konstruksi atau pembangunan kerangka atau desain awal dari perangkat lunak yang akan dikembangkan.



Gambar 1. Alur Metode *Prototyping*

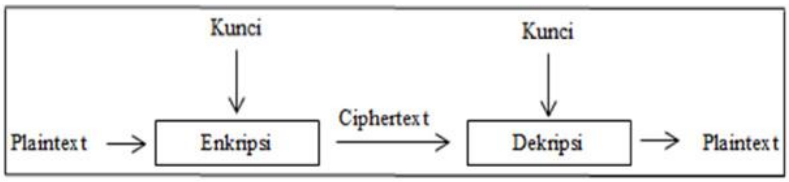
p-ISSN: 2301-5373
e-ISSN: 2654-5101

2.1. MAC Address

MAC *Address* (*Media Access Control*) adalah alamat perangkat keras yang unik yang diberikan kepada *network interface* oleh produsen. Ini unik untuk setiap *network interface*. *Network interface* bisa berupa *Network Interface Card* (NIC) untuk jaringan kabel atau dapat berupa adaptor nirkabel. Ada berbagai perusahaan yang memproduksi *Network Interface* dan mereka memasukkan alamat perangkat keras unik di setiap *Network Interface*. Alamat ini secara unik mengidentifikasi sebuah antarmuka jaringan. Alamat Mac adalah alamat 48-bit yang direpresentasikan dalam sistem heksadesimal. 24 bit pertama mewakili kode khusus perusahaan yang diberikan oleh IEEE kepada setiap perusahaan manufaktur. Misalnya, 00:13:10 adalah kode (Organisasi Pengidentifikasi Unik) yang terkait dengan Linksys. Ada lebih dari satu kode yang terkait dengan satu perusahaan juga. Dan 24 bit berikutnya mewakili nomor khusus antarmuka untuk setiap antarmuka jaringan/adaptor tunggal.

2.2. Kriptografi

Kriptografi adalah seni dan ilmu untuk menjaga keamanan pesan dengan mengubahnya menjadi kode-kode yang hanya dapat dipahami oleh pihak yang dituju.Gambaran proses dapat dilihat pada Gambar 2 Proses ini, disebut enkripsi, mengubah data asli (*plaintext*) menjadi bentuk yang tidak dapat dikenali (*ciphertext*) untuk melindunginya dari pihak ketiga yang tidak diinginkan. Setelah pesan tersebut diterima, penerima melakukan dekripsi untuk mengembalikan data ke bentuk semula agar dapat dimengerti. Proses enkripsi dan dekripsi memerlukan kunci, yang biasanya berupa string atau deretan bilangan, untuk memastikan keamanan dan integritas pesan.



Gambar 2. Proses Enkripsi dan Dekripsi

2.3. Digital Signature

Digital signature atau *electronic signature* adalah berkas yang memungkinkan untuk membuktikan pada pihak ketiga bahwa suatu dokumen tertentu telah disetujui oleh suatu entitas (misalnya, individu atau perusahaan) seperti halnya tanda tangan kertas. Oleh karena itu, tanda tangan digital merupakan mekanisme pengamanan yang dapat diandalkan [4]. Tanda tangan digital memiliki karakteristik sebagai berikut:

a. *Authenticity* : identitas dari pengirim harus dapat ditemukan dengan pasti.
b. *Non forgery* : tanda tangan tidak dapat dipalsukan. Seseorang tidak dapat berpura-pura menjadi orang lain.
c. *Non reusable* : tanda tangan tidak dapat digunakan ulang. Ia merupakan bagian dari dokumen yang ditandatangani dan tidak dapat dipindahkan ke dokumen lain.
d. *Inalterability* : dokumen yang telah ditandatangani tidak dapat diubah. Setelah ditandatangani, ia tidak dapat diubah.
e. *Irrevocability* : orang yang menandatangani dokumen tidak dapat menyangkalnya.

Tanda tangan digital memberikan autentikasi identitas, mendeteksi manipulasi data yang tidak sah, menjamin keutuhan data, dan merupakan satu-satunya cara untuk memastikan non-repudiasi di dunia digital. Non-repudiasi memberikan bukti kepada pihak ketiga oleh pihak yang menandatangani. Setelah itu, pihak yang menandatangani tidak dapat menyangkal kegiatan dengan pihak ketiga atau menolak tanda tangan tersebut [5].

2.4. SHA (Secure Hash Algorithm)

SHA-256 adalah sebuah fungsi hash yang menggunakan ukuran digest 256-bit pada versi SHA-2. Fungsi hash ini didasarkan pada MD4 yang dikembangkan oleh Ronald L. Rivest dari MIT. SHA-256 menggunakan enam operasi logika dasar seperti AND, OR, XOR, *shift right*, dan *rotate right*. Algoritma ini mengubah sebuah *message schedule* yang terdiri dari 64 elemen 32-bit *word*, delapan variabel 32-bit, dan variabel penyimpan nilai hash sebanyak delapan word 32-bit. Pesan masukan diubah menjadi message digest sepanjang 256-bit dengan menggunakan algoritma ini.

Menurut *Secure Hash Signature Standard*, jika panjang pesan masukan kurang dari 2^64, maka harus dioperasikan dalam kelompok 512-bit dan diubah menjadi sebuah message digest sepanjang 256-bit.

2.5. Algoritma AES

Advanced Encryption Standard (AES) merupakan salah satu algoritma kriptografi yang digunakan untuk mengamankan data. Algoritma AES dapat mengenkripsi data menjadi sebuah *chippertext* yang tidak dapat dibaca langsung. Untuk melihat data yang terenkripsi kita perlu mendekripsi data tersebut. Proses dekripsi ini akan mengembalikan *chipertext* menjadi data awal atau *plaintext*. Algoritma AES dapat mengenkripsi dan mendekripsi data dengan Panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit. Perbedaan dari ketiga urutan ini terletak pada panjang kunci yang mempengaruhi jumlah perputaran (*round*) yang dapat digambarkan dalam bentuk Tabel 1 berikut:

Tabel 1. Algoritma AES

	Panjang kunci	Panjang blok	Jumlah putaran
AES-128	4	4	10
AES-129	6	4	12
AES-256	8	4	14

Pada Tabel 1 dapat dilihat panjang dari masing-masing kunci, blok dan jumlah putaran dari setiap algoritma. Pada penelitian ini menggunakan AES-128 bit dengan 10 kali jumlah putaran. Terdapat 4 tranformasi putaran pada proses enkripsi dan dekripsi [6]:

a. SubBytes

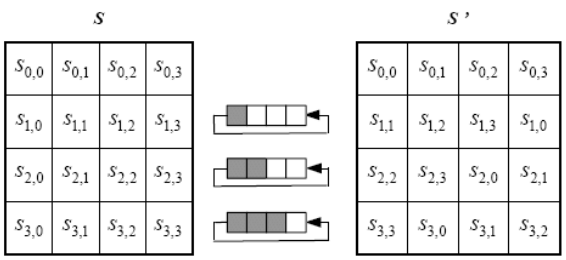
Proses menggunakan tabel substitusi untuk menukar isi byte. Penukaan per bit ini dilakukan dengan mencocokkan nilai bit dengan s box seperti yang terlihat pada Gambar 3.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Gambar 3. S-Box

b. ShiftRows

Pada langkah ini dilakukan pergeseran blok per baris pada state array. Dapat dilihat pada Gambar 4 *ShiftRows* dilakukan dengan perhitungan sebagai berikut: baris pertama tidak mendapatkan pergeseran, baris kedua mengalami pergeseran ke kiri sebanyak satu kali, baris ketiga mengalami pergeseran sebanyak dua kali, dan baris keempat mengalami pergeseran sebanyak tiga kali.



Gambar 4. Shift Row

c. MixColumn

Proses mengalikan blok data atau pengacakan di masing-masing *state array* dengan rumus 2.1 sebagai berikut:

p-ISSN: 2301-5373
e-ISSN: 2654-5101

$$A(x) = \{03\}x^2 + \{01\}x^2 + \{01\}x + \{02\} \quad (2,1)$$

d. AddRoundKey
Menggabungkan *round key* dan *round key* dengan hubungan XOR.

Proses dekripsi algoritma AES:

1) InvShiftRows
Melakukan pergeseran bit ke kanan pada setiap blok baris.
2) InvSubBytes
Element state dipetakan dengan tabel inverse S-box
3) InvMixColumn
Mengalikan kolom *state* dengan matriks AES.
4) AddRoundKey
Menggabungkan *round key* dan *round key* dengan hubungan XOR.

Algoritma AES ini merupakan array of bytes dengan dua dimensi yang disebut dengan state. Rumus yang digunakan dalam state adalah NROW * NCOLS. Input bytes digunakan untuk menyimpan enkripsi data yang kemudian di salin kedalam array state, dan hasil keluaran atau hasil dari enkripsi dan dekrispi akan disimpan ke dalam output bytes. Proses awal enkrisi adalah transformasi AddRoundKey yang terjadi pada salinan input yang disimpan didalam state, lalu state akan mengalami beberapa transformasi yaitu SubBytes, ShiftRows, MixColumns, dan AddRoundKey secara berulang sebanyak 9 putaran/round. Untuk round terakhir atau round 10, state tidak mengalami transformasi MixColumns [6].

3. Hasil dan Pembahasan

Pengimplementasian sistem *digital signature* untuk verifikasi dokumen digital menggunakan MAC *address* pada SHA-256 dan AES-128 menggunakan perangkat keras dan perangkat lunak. Sistem ini dibangun menggunakan bahasa pemrograman GO. Berikut merupakan spesifikasi dari perangkat keras yang penulis gunakan :

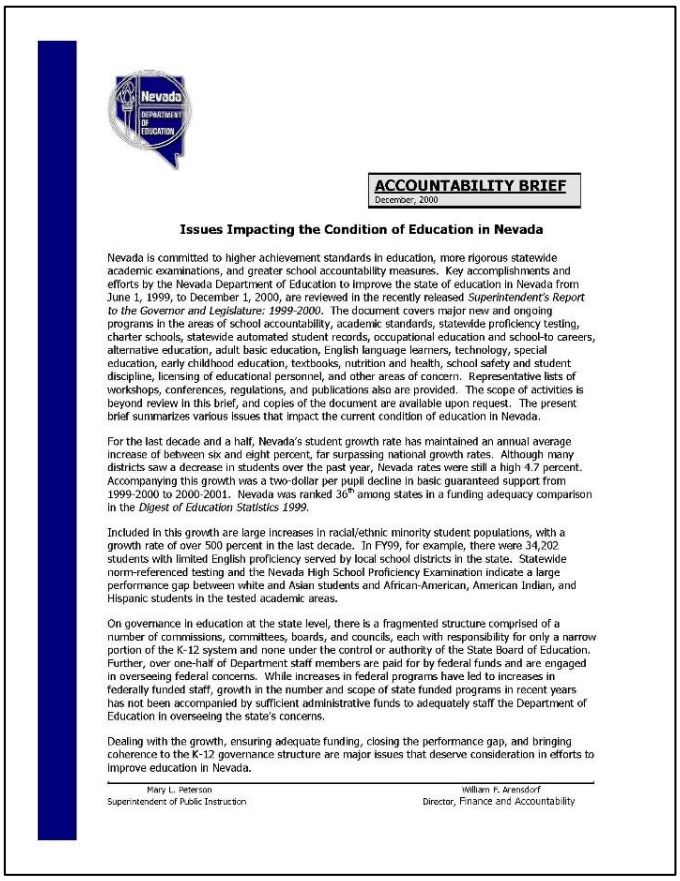
1. Processor AMD Ryzen 5 3,50 Ghz
2. RAM 16 GB
3. SSD 512 GB

Berikut perangkat lunak yang penulis gunakan :

1. Visual Studio Code
2. Sistem Operasi Windows 11

3.1. Implementasi Algoritma AES128 dan SHA256

Algoritma AES128 pada aplikasi ini diimplementasikan pada fitur *Sign Document* dan *Verify Document* sedangkan algoritma SHA256 diimplementasikan pada fitur *Sign Document*. Fitur *Sign Document* menggunakan metode enkripsi AES128 dan SHA256 serta pada fitur *Verify Document* menggunakan metode dekripsi AES128. User akan diminta untuk menginputkan dokumen yang akan ditandatangani beserta id user tujuan yang akan menerima kunci, dokumen yang diinputkan ini yang akan dienkripsi dan disimpan kedalam database serta kunci dari enkripsi tersebut akan dikirimkan ke user sesuai dengan id user yang diinputkan pada saat melakukan proses enkripsi. Pada proses pembentukan kunci membutuhkan MAC *address* dari komputer user, kemudian MAC *address* tersebut akan ditambahkan random bit oleh sistem hingga menjadi 16 bit key yang akan digunakan pada proses enkripsi dan dekripsi. Kemudian user yang memiliki kunci dan ingin melakukan verifikasi dokumen dapat melakukannya dengan menginputkan dokumen beserta kunci yang telah diberikan pada menu *Verify Document*. Pada proses *verify* inilah menggunakan metode dekripsi. Contoh pada Gambar 5. Pada menu *Sign Document* pengguna perlu memasukkan user ID (receiver_id) yang akan menjadi penerima kunci, pengguna yang ingin menjadi penerima kunci harus sudah terdaftar didalam sistem terlebih dahulu, sedangkan untuk mac_address akan dimasukkan secara otomatis oleh sistem. Untuk file *document* yang akan dimasukkan haruslah dalam format .pdf, jika pengguna memasukkan file *document* selain menggunakan format .pdf maka sistem akan memberikan notifikasi format dokumen tidak valid.



ACCOUNTABILITY BRIEF
December, 2000

Issues Impacting the Condition of Education in Nevada

Nevada is committed to higher achievement standards in education, more rigorous statewide academic examinations, and greater school accountability measures. Key accomplishments and efforts by the Nevada Department of Education to improve the state of education in Nevada from June 1, 1999, to December 1, 2000, are reviewed in the recently released *Superintendent's Report to the Governor and Legislature: 1999-2000*. The document covers major new and ongoing programs in the areas of school accountability, academic standards, statewide proficiency testing, charter schools, statewide automated student records, occupational education and school-to careers, alternative education, adult basic education, English language learners, technology, special education, early childhood education, textbooks, nutrition and health, school safety and student discipline, licensing of educational personnel, and other areas of concern. Representative lists of workshops, conferences, regulations, and publications also are provided. The scope of activities is beyond review in this brief, and copies of the document are available upon request. The present brief summarizes various issues that impact the current condition of education in Nevada.

For the last decade and a half, Nevada's student growth rate has maintained an annual average increase of between six and eight percent, far surpassing national growth rates. Although many districts saw a decrease in students over the past year, Nevada rates were still a high 4.7 percent. Accompanying this growth was a two-dollar per pupil decline in basic guaranteed support from 1999-2000 to 2000-2001. Nevada was ranked 36th among states in a funding adequacy comparison in the *Digest of Education Statistics 1999*.

Included in this growth are large increases in racial/ethnic minority student populations, with a growth rate of over 500 percent in the last decade. In FY99, for example, there were 34,202 students with limited English proficiency served by local school districts in the state. Statewide norm-referenced testing and the Nevada High School Proficiency Examination indicate a large performance gap between white and Asian students and African-American, American Indian, and Hispanic students in the tested academic areas.

On governance in education at the state level, there is a fragmented structure comprised of a number of commissions, committees, boards, and councils, each with responsibility for only a narrow portion of the K-12 system and none under the control or authority of the State Board of Education. Further, over one-half of Department staff members are paid for by federal funds and are engaged in overseeing federal concerns. While increases in federal programs have led to increases in federally funded staff, growth in the number and scope of state funded programs in recent years has not been accompanied by sufficient administrative funds to adequately staff the Department of Education in overseeing the state's concerns.

Dealing with the growth, ensuring adequate funding, closing the performance gap, and bringing coherence to the K-12 governance structure are major issues that deserve consideration in efforts to improve education in Nevada.

Mary L. Peterson
Superintendent of Public Instruction

William P. Arensdorf
Director, Finance and Accountability

Gambar 5. Filetest7.pdf

Ketika pengguna melakukan proses sign document, sistem akan melakukan proses enkripsi dengan tahapan sebagai berikut :

a. *Generate 16-byte key*, proses diawali dengan sistem membuat 16 byte key dengan menggunakan kombinasi 6 byte mac address yang telah dimasukkan pengguna E0-70-EA-AC-C5-AD dan 10 random byte. Kombinasi kunci yang didapatkan menggunakan mac address E0-70-EA-AC-C5-AD dalam bentuk hex adalah [E070EAACC5ADC310C34A84B7356E2713].

b. Merubah bentuk file menjadi byte, setelah mendapatkan kunci untuk melakukan proses enkripsi, maka proses selanjutnya adalah merubah bentuk file menjadi byte untuk selanjutnya bisa dilakukan proses enkripsi. Berikut pada Tabel 2 merupakan contoh bentuk hex dari byte yang telah dirubah :

Tabel 2. *File Document Converted To Hex*

255044462d312e32200d25e2e3cfd3200d31302030206f626a0d3c3c0d2f4c656e67746820313120302052 0d2f46696c746572202f466c6174654654656 36f6465200d3e3e0d73747265616d0d0a4889d4574b8f14d715fe05fd1feed2917a7aaaaadfd9cdd38c4d6cc24c842c6573bbea76f735f568ea31cde41720b1882564458a712484104618826290e50814b14059390b62b18058ac501c4b281b1e822025dfb98fee1aaa31c459a58d7b66aaee3ddf39dffdcee33a0dafcda6ace6b17758cd6523566b3aacdb711add3673e89dfa4a456dc86a4ea3d7efa9254d8f9e769c1e2d73fb4dd6741b76d59b1be9624fc77399dbf3b49df9aa1f61a4db5246ba66c9ea4e6d79138f5b6c675873ecda0eb9ede1d93a73d88e4

c. Merubah kedalam bentuk string, setelah file dirubah kedalam bentuk *bytes*, agar bisa dienkripsi menggunakan algoritma SHA256, file yang telah berbentuk bytes akan diubah terlebih dahulu kedalam bentuk string menggunakan algoritma base64. Kemudian menambahkan MAC *address* keladalam

p-ISSN: 2301-5373
e-ISSN: 2654-5101

string sebagai *digital signature*. Contoh bentuk dari *bytes* yang telah dirubah menjadi string dapat dilihat pada Tabel 3 :

Tabel 3. *Byte Converted To String*

JVBERi0xLjIgDSXi48/TIA0xMCAwIG9iag08PA0vTGVuZ3RoIDExIDAgUg0vRmlsdGVyIC9GbGF0ZURlY29kZSANPj4Nc3RyZWFtDQpIidRXS48U1xX+Bf0f7tKRenqqqt/ZzdOMTWzCTIQsZXO76nb3NfVo6jHN5BcgsYglZEWKcSSEEEYYgmKQ5QgUsUBZOQtisYBYrFAcSygbHoIgJd+5j+4aqjHEWaWNe2aq7j3fOd/9zuM6Da/NpqzmsXdYzWUjVms6rNtxGt02c+id+kpFbchqTqPX76klTY+edpweLXP7TdZ0G3bVmxvpYk/Hc5nb87Sd+aofYaTbUka6ZsnqTm15E49bbGdYc+zaDrnt4dk6c9iOT19T9tbK2tr7v3hvZ2V16+DWzgds9fDWxuZPdj6sbezMAbRxjTF3xDLR7FsmnEblXa/67r+MzxBtePJ+NE+ucr3do2/lcYkrj/WqVLlNWgq23l//

d. Menambahkan MAC *address* kedalam string sebagai digital signature. Setelah formal file diubah menjadi string, MAC *address* yang berupa *byte* akan dirubah menjadi bentuk string [4HDqrMWt]. Kemudian string MAC *address* tersebut akan ditambahkan pada bagian awal dari string file document seperti pada Tabel 4.

Tabel 4. *Add Digital Signature*

4HDqrMWtJVBERi0xLjIgDSXi48/TIA0xMCAwIG9iag08PA0vTGVuZ3RoIDExIDAgUg0vRmlsdGVyIC9GbGF0ZURlY29kZSANPj4Nc3RyZWFtDQpIidRXS48U1xX+Bf0f7tKRenqqqt/ZzdOMTWzCTIQsZXO76nb3NfVo6jHN5BcgsYglZEWKcSSEEEYYgmKQ5QgUsUBZOQtisYBYrFAcSygbHoIgJd+5j+4aqjHEWaWNe2aq7j3fOd/9zuM6Da/NpqzmsXdYzWUjVms6rNtxGt02c+id+kpFbchqTqPX76klTY+edpweLXP7TdZ0G3bVmxvpYk/Hc5nb87Sd+aofYaTbUka6ZsnqTm15E49bbGdYc+zaDrnt4dk6c9iOT19T9tbK2tr7v3hvZ2V16+DWzgds9fDWxuZPdj6sbezMAbRxjTF3xDLR7FsmnEblXa/67r+MzxBtePJ+NE+ucr3do2/lcYkrj/WqVLlNWgq23l//

e. Enkripsi SHA256, setelah mengubah *file* berbentuk *byte* kedalam string, kemudian string tersebut akan dienkripsi menggunakan algoritma SHA256 untuk mencari *plaintext* yang akan digunakan kedalam enkripsi AES128. Hasil dari enkripsi SHA256 dapat dilihat pada Tabel 5 :

Tabel 5. *Plaintext*

d403aa66af446619701d159ea80a2dcf6b179bdda0ef821262f9df925d1e30e9

f. Hasil enkripsi dari *plaintext* yang didapatkan seperti pada Tabel 6:

Tabel 6. *Ciphertext*

8ukN4wqblVHCOFqVNRThs2RqCSZnmcwI9Nd71hYSnFZcfQgKc42hc99XtoCeNdpt3K2/LH6RDu1elLbQqnxZPvnuluEpLk5i2AyOF5FjrYw=

Setelah mendapatkan hasil dari enkripsi *file document*, hasil enkripsi tersebut akan disimpan kedalam database. Serta kunci akan langsung bisa diakses oleh pengguna yang dituju sesaat setelah sign document berhasil. Untuk melakukan verifikasi dokumen dapat menggunakan menu *Verify Document*. Kunci yang telah dibagikan dapat dilihat dan dipilih pada menu *Verify Document*. Pada menu *Verify Document* disediakan dropdown tempat pengguna dapat melihat kunci yang telah dibagikan kepada pengguna tersebut, yang nantinya akan berguna untuk melakukan proses verifikasi dokumen.

Setelah pengguna memilih kunci yang akan digunakan dalam proses verifikasi dokumen, user akan diminta memasukkan file yang akan diverifikasi. Kemudian ketika pengguna menekan tombol pilih, sistem akan langsung mengubah bentuk *file document* yang dimasukkan oleh user menjadi *byte* dan setelah itu mengubah *byte* menjadi string sama seperti pada proses enkripsi. Setelah file document

menjadi string kemudian akan dienkripsi dengan algoritma SHA256. Setelah mendapatkan hasil enkripsi dari file masukkan dari pengguna, sistem akan melakukan dekripsi pada ciphertext file yang akan diverifikasi. Dengan mencocokkan *Sign ID* yang terdapat pada *table keys,* sistem akan mendapatkan *ciphertext*, kemudian akan dilanjutkan dengan melakukan dekripsi menggunakan algoritma AES128. Melalui proses dekripsi tersebut akan didapatkan string SHA256. Kemudian string SHA256 yang didapatkan dari enkripsi file yang dimasukkan oleh pengguna akan dibandingkan dengan string SHA256 yang telah didekripsi oleh sistem. Jika hasil perbandingan tersebut sama maka dokumen yang dimasukkan oleh pengguna tersebut adalah dokumen yang asli. akan dienkripsi.

Pengujian *avalanche effect* dilakukan untuk menilai seberapa baik sistem enkripsi dapat mengubah perubahan kecil dalam input menjadi perubahan besar dan acak dalam outputnya. Dengan memodifikasi setiap bit input, pengujian ini mengamati seberapa signifikan perubahan output. Algoritma kriptografi yang baik harus selalu memenuhi nilai persamaan Avalanche > 50%. Hal ini memastikan bahwa penyerang tidak dapat dengan mudah memprediksi ciphertext dari teks asli atau sebaliknya.

Nama File	Halaman File	Ukuran File	Berisi Gambar	Rata – rata Avalanche effect
Filetest1.pdf	1	19 KB	Tidak	52.47%
Filetest2.pdf	5	25 KB	Tidak	51.17%
Filetest3.pdf	1	26 KB	Ya	50.26%
Filetest4.pdf	4	30 KB	Ya	50.13%
Filetest5.pdf	2	31 KB	Ya	50.39%
Filetest6.pdf	1	33 KB	Ya	53.12%
Filetest7.pdf	1	40 KB	Ya	49.74%
Filetest8.pdf	1	56 KB	Ya	51.95%
Filetest9.pdf	6	62 KB	Ya	49.48%
Filetest10.pdf	7	165 KB	Ya	50.39%
Filetest11.pdf	2	234 KB	Ya	49.51%
Filetest12.pdf	2	432 KB	Ya	52.23%
Filetest13.pdf	4	666 KB	Ya	53.75%
Filetest14.pdf	3	924 KB	Ya	50.32%
Filetest15.pdf	2	1,084 KB	Ya	49.92%
Filetest16.pdf	1	1,347 KB	Ya	51.02%
Filetest17.pdf	1	1,706 KB	Ya	50.12%
Filetest18.pdf	2	1,789 KB	Ya	52.64%
Filetest19.pdf	1	2,636 KB	Ya	49.42%
Filetest20.pdf	8	3,668 KB	Ya	50.84%

Algoritma kriptografi yang tidak memenuhi persamaan efek avalanche ini mudah ditembus oleh kriptanalisis [7].

Tabel 7. Pengujian *Avalanche Effect*

Hasil Tabel 7 didapatkan dari hasil pengujian *avalanche effect* dari *file document*. Langkah melakukan pengujian *avalanche effect* adalah dengan merubah satu bit dari *plaintext* atau dalam penelitian ini yaitu kode *file document*. Selanjutnya dilakukan perbandingan bit *ciphertext* dari *plaintext* pertama dan bit *ciphertext* dari *plaintext* yang dirubah. Setelah mendapatkan hasil perbandingan berupa jumlah bit yang berbeda dari kedua *ciphertext*, selanjutnya jumlah bit yang berbeda dibangi dengan panjang ciphertext dikalikan dengan 100%. Percobaan dilakukan sebanyak 20 kali, dengan mengubah satu buah karakter acak dari hasil hashing *file document* didapatkan rata-rata *avalanche effect* 50.94%. Yang dimana dalam studi literatur mengenai avalanche effect dijelaskan bahwa sistem avalanche effect diatas 50% sudah dapat dikategorikan sebagai sistem yang baik, karena sudah mengakibatkan masalah yang cukup sulit untuk dipecahkan oleh para penyerang [7].

Tabel 8. Pengujian *Black-Box*

No	Skenario	Deskripsi	Hasil yang diharapkan	Hasil pengujian
1	Login	Login dengan username dan password yang terdaftar	Pengguna dapat masuk kedalam sistem	Pengguna dapat masuk kedalam sistem

p-ISSN: 2301-5373
e-ISSN: 2654-5101

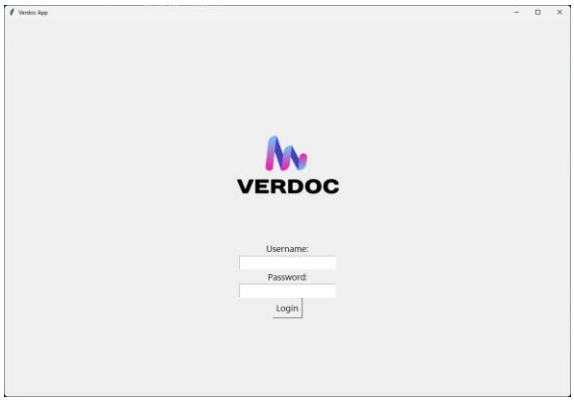
		Login dengan username atau password yang salah	Pengguna tidak dapat masuk kedalam sistem dan menadapatkan penanda bahwa username atau password salah.	Pengguna tidak dapat masuk kedalam sistem dan menadapatkan penanda bahwa username atau password salah.
2	Registrasi	Pengguna melakukan regristrasi	Pengguna berhasil melakukan registrasi	Pengguna berhasil melakukan registrasi
3	Melakukan *Sign Document*	Pengguna memasukkan *file document* beserta *receiver id* yang tersedia pada dropdown user	Dokumen berhasil ditandatangani dan disimpan kedalam database	Dokumen berhasil ditandatangani dan disimpan kedalam database
4	Melakukan *Verify Document*	Pengguna memasukkan *file document* yang benar beserta *key* yang benar dan tersedia pada dropdown *keys*	Dokumen berhasil diverifikasi keasliannya	Dokumen berhasil diverifikasi keasliannya
		Pengguna memasukkan *file document* yang salah beserta *key* yang salah dan tersedia pada dropdown *keys*	Dokumen tidak berhasil diverifikasi keasliannya	Dokumen tidak berhasil diverifikasi keasliannya

Hasil pengujian *Black-Box* memastikan bahwa setiap fungsi utama dalam sistem berjalan sesuai dengan spesifikasi dan mampu menangani berbagai skenario yang mungkin terjadi dalam penggunaan nyata. Dari ke 4 skenario pengujian yang telah dijalankan, aplikasi dapat berjalan dengan baik. Pada skenario login, sistem berhasil memverifikasi kredensial pengguna dengan benar, memastikan bahwa hanya pengguna yang sah yang dapat mengakses aplikasi. Skenario registrasi menunjukkan bahwa sistem mampu menangani pendaftaran pengguna baru dengan tepat, menyimpan data yang diperlukan dalam database. Skenario *sign document* menunjukkan bahwa sistem mampu memproses penandatanganan dokumen elektronik dengan benar, memastikan integritas dan keaslian dokumen. Terakhir, skenario *verify document* memastikan bahwa sistem dapat memverifikasi keabsahan dokumen yang telah ditandatangani, dengan memeriksa hasil hash dekripsi dan membandingkannya dengan hash yang diinputkan. Keberhasilan dalam keempat skenario ini menunjukkan bahwa sistem aplikasi berfungsi sesuai dengan spesifikasi dan mampu menangani operasi-operasi inti dengan baik.

3.2. Tampilan Antarmuka Sistem

Bagian utama terdiri dari beberapa menu yang dapat diakses oleh user sesuai dengan fungsinya masing – masing. Berikut merupakan tampilan beserta keterangan dari menu tersebut :

1. Menu Home
   Menu ini merupakan menu utama ketika user berhasil login kedalam aplikasi.
2. Menu Sign Document
   Menu ini berfungsi untuk melakukan enkripsi terhadap document
3. Menu Verify Document
   Menu ini berfungsi untuk melakukan dekripsi terhadap document



Gambar 6. Tampilan Antarmuka Sistem

4. Kesimpulan

Berdasarkan hasil yang telah didapatkan dari penelitian ini, maka dapat disimpulkan bahwa perancangan aplikasi Verdoc yang mengimplementasikan digital signature menggunakan mac address pada SHA256 dan AES128 mampu mengimplementasikan algoritma kriptografi AES128 dalam mengamankan keaslian document menggunakan digital signature. Dari hasil penelitian yang dilakukan dapat dilihat bahwa sistem dapat melakukan proses hashing dan enkripsi dengan sangat baik dimulai dari proses pembuatan 16 Byte kunci menggunakan mac address. Setelah sistem melakukan proses hashing dan enkripsi terhadap document, sistem dapat menyimpan hasil enkripsi tersebut kedalam database dan memberikan hak akses kepada penerima kunci. Sistem juga dapat melakukan proses dekripsi secara sangat baik dapat dibuktikan dengan berhasil mendekripsi kembali ciphertext berupa dokumen hasil hashing yang telah disimpan didalam database dan mendapatkan hasil yang sama dengan document asli yang diverifikasi oleh pengguna. Hasil pengujian Black-Box menunjukkan bahwa keempat skenario utama, yaitu login, registrasi, sign document, dan verify document, berhasil dijalankan dengan baik. Pada skenario login, sistem berhasil memverifikasi kredensial pengguna dengan benar, memastikan akses hanya untuk pengguna yang sah. Skenario registrasi menunjukkan bahwa sistem mampu menangani pendaftaran pengguna baru dengan tepat dan menyimpan data yang diperlukan dalam database. Skenario sign document menunjukkan bahwa sistem mampu memproses penandatanganan dokumen elektronik dengan benar, menjaga integritas dan keaslian dokumen. Terakhir, skenario verify document memastikan bahwa sistem dapat memverifikasi keabsahan dokumen yang telah ditandatangani dengan memeriksa hasil hash dekripsi dan membandingkannya dengan hash yang diinputkan. Keberhasilan dalam keempat skenario ini menunjukkan bahwa sistem aplikasi berfungsi sesuai dengan spesifikasi dan mampu menangani operasi-operasi inti dengan baik.

Performa dan akurasi sistem dalam melakukan enkripsi dapat dikatakan mendapatkan kualitas baik yang dimana didapatkan berdasarkan data hasil pengujuan *avalance effect* yang dilakukan pada file document yang diinputkan oleh pengguna. Dalam pengujian ini file document yang digunakan adalah Filetest7.pdf. Percobaan dilakukan sebanyak 20 kali, dengan mengubah satu buah karakter acak dari hasil hashing file document didapatkan rata-rata avalanche effect 50.94%. Yang dimana dalam studi literatur mengenai avalanche effect dijelaskan bahwa sistem avalanche effect diatas 50% sudah dapat dikategorikan sebagai sistem yang baik, karena sudah mengakibatkan masalah yang cukup sulit untuk dipecahkan oleh para penyerang.

Referensi

[1] Genc, Y., & Afacan, E. (2021). Design and implementation of an efficient elliptic curve digital signature algorithm (ECDSA). 2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 - Proceedings. https://doi.org/10.1109/IEMTRONICS52119.2021.9422589

[2] Suhandinata, S., Rizal, R. A., Wijaya, D. O., Warren, P., & Srinjiwi, S. (2019). Analisis Performa Kriptografi Hybrid Algoritma Blowfish Dan Algoritma Rsa. JURTEKSI (Jurnal Teknologi Dan Sistem Informasi), 6(1), 1–10. https://doi.org/10.33330/jurteksi.v6i1.395

[3] Simbolon, I. A. R., Gunawan, I., Kirana, I. O., Dewi, R., & Solikhun, S. (2020). Penerapan Algoritma AES 128-Bit dalam Pengamanan Data Kependudukan pada Dinas Dukcapil Kota Pematangsiantar. Journal of Computer System and Informatics (JoSYC), 1(2), 54–60

[4] Josias Gbetoho Saho, N., & Ezin, E. C. (2019). Securing Document by Digital Signature through RSA and Elliptic Curve Cryptosystems. 2019 International Conference on Smart Applications, Communications and Networking, SmartNets 2019. https://doi.org/10.1109/SmartNets48225.2019.9069749

[5] Boneh, D. (2011). Digital Signature Standard. Encyclopedia of Cryptography and Security, 347–347. https://doi.org/10.1007/978-1-4419-5906-5_145

[6] Prameshwari, A., & Sastra, N. P. (2018). Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen. Eksplora Informatika, 8(1), 52. https://doi.org/10.30864/eksplora.v8i1.139

[7] Verma, R., & Sharma, A. K. (2020). Cryptography: Avalanche effect of AES and RSA. International Journal of Scientific and Research Publications (IJSRP), 10(4), p10013. https://doi.org/10.29322/ijsrp.10.04.2020.p10013